Análisis de sentimientos del mercado financiero utilizando procesamiento de lenguaje natural en Twitter

Sheryl Alcántara Rojas[1], Daniel Alfaro Figueroa [1], Kevin Ibarra Valverde[1], Luis Fernando Solano Coto [1]
sheryl.alcantara@ucr.ac.cr, daniel.alfarofigueroa@ucr.ac.cr, kevin.ibarra@ucr.ac.cr.
luis.solanocoto@ucr.ac.cr

RESUMEN

El artículo explora el uso del procesamiento de lenguaje natural (NLP) para analizar los sentimientos expresados en la red social Twitter, con el objetivo de evaluar la percepción del mercado financiero. Se exploran las metodologías más utilizadas en este campo y se examinan sus características y enfoques. Se construyen tres modelos de redes neuronales para la clasificación de los tuits según su sentimiento (alcista, neutral o bajista): una red neuronal recurrente (RNN), una red neuronal convolucional (CNN) y un modelo de perceptrón multicapa (MLP). Los modelos se entrenan con un conjunto de datos de Twitter Financial News, y se evalúan utilizando la métrica F1 ponderado. Los resultados muestran que el modelo de mejor rendimiento es el que utiliza una red neuronal recurrente (RNN).

PALABRAS CLAVE: aprendizaje, clasificación, twitter, finanzas

INTRODUCCIÓN

El análisis de sentimientos permite identificar la percepción de los usuarios en diversos aspectos de la vida diaria, como sus preferencias por productos en el mercado, el nivel de confianza en entornos laborales, políticas o como en este estudio en el área financiera; se analizan los sentimientos expresados en una red social. El objetivo principal es anticipar tendencias o elecciones basándose en las emociones expresadas. En este artículo, exploramos las metodologías más utilizadas en este campo, examinando sus características y enfoques.

Los usuarios de Twitter hacen uso de la plataforma para expresar desde opiniones hasta emociones sobre cualquier tópico. Los modelos de clasificación inteligentes han demostrado su capacidad de predicción de sentimientos en textos, para determinar la percepción de los usuarios sobre aspectos de la vida cotidiana (Mostafa, 2013).

Las opiniones en línea se han analizado recientemente utilizando el análisis de sentimientos (SA, por sus siglas en inglés). Esto es, básicamente, una aplicación del procesamiento del lenguaje natural (NLP) que emplea la lingüística computacional y la minería de texto para identificar el sentimiento de un texto, categorizando típicamente como positivo, neutral o negativo. Esta técnica también es conocida en la literatura de minería de texto como análisis de polaridad emocional (EPA), minería de opiniones, minería de reseñas o extracción de valoraciones (Zagal, Tomuro, & Shepitsen, 2012). Por lo tanto, el SA puede considerarse

[1] Estudiante del Bachillerato en Estadística de la Facultad de Ciencias Económicas, Universidad de Costa Rica, San Pedro de Montes de Oca, San José, Costa Rica.

como una técnica automatizada de descubrimiento de conocimiento que tiene como objetivo encontrar patrones ocultos en un gran número de reseñas, blogs o tweets.

Estos datos temporales que cuantifican las emociones del público general hacia compañías que cotizan en bolsa, personalidades y eventos del mundo de las finanzas, son utilizados por muchos inversores como información social en sus estrategias de compra y venta en los mercados financieros. La extensa investigación en finanzas conductuales ha mostrado evidencia del hecho que los inversores reaccionan a las noticias, y que, por lo general, muestran una mayor propensión a hacer un movimiento de inversión basado en malas noticias en lugar de buenas noticias (Chan, 2003).

Si bien no existe un consenso general sobre la posibilidad de predecir los movimientos en los precios de acciones en los mercados financieros mediante el análisis del sentimiento de las noticias (Schoen et al., 2013), el hecho relevante es que esta área de investigación es actualmente una de las más populares en el desarrollo de modelos de pronóstico que hacen uso de grandes masas de datos, tanto a nivel académico como industrial.

METODOLOGÍA

El conjunto de datos Twitter Financial News es un corpus en inglés de tuits relacionados con finanzas. Este se utiliza para clasificar los tuits financieros según su sentimiento (Hugging Face, 2022). El conjunto contiene 11,931 entradas etiquetadas con 3 categorías: 1 es Alcista (en inglés, *bullish*): Indica una perspectiva positiva o de crecimiento, 2 es Neutral, y 0 es Bajista (en inglés, *bearish*): Indica una perspectiva negativa o de caída. Los datos fueron recolectados utilizando la API de Twitter. Este conjunto de datos es compatible con tareas de clasificación multiclase.

Los datos venían divididos previamente en un conjunto de entrenamiento y validación, los cuales contaban con 9543 y 2388 tuits respectivamente. Al tener más datos en el conjunto de entrenamiento se decidió extraer de aquí el 20% de los datos para conformar el conjunto de test, dejando así 1909 tuits en este conjunto. Posterior a esta división, los datos de texto fueron procesados utilizando un encoder para convertirlos en secuencias numéricas. Finalmente, se aplicaron técnicas de aumentación de datos al conjunto de datos de entrenamiento para mitigar el posible efecto del desbalance de clases que se puede observar en la figura 2. Estas técnicas incluyeron el reemplazo aleatorio de palabras por sinónimos y la eliminación o inserción de palabras en las secuencias para generar nuevas muestras. Estas técnicas fueron adaptadas del repositorio de GitHub de Esrat Maria (2020), donde se exploran métodos de aumento de datos para procesamiento de lenguaje natural. La distribución de los datos después de la aplicación de las técnicas se puede visualizar en la figura 1.

Figura 1

Distribución de los valores de la columna “label” para los datos del conjunto de entrenamiento aumentados

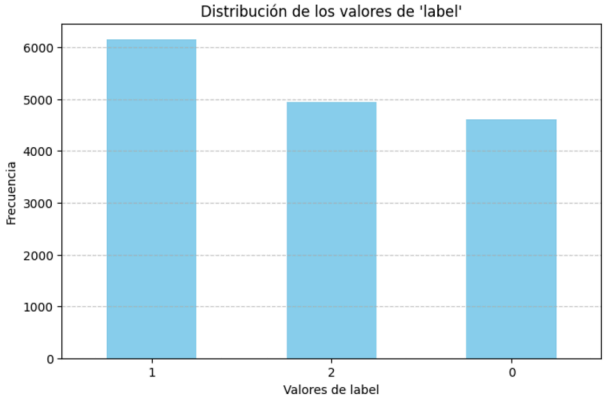
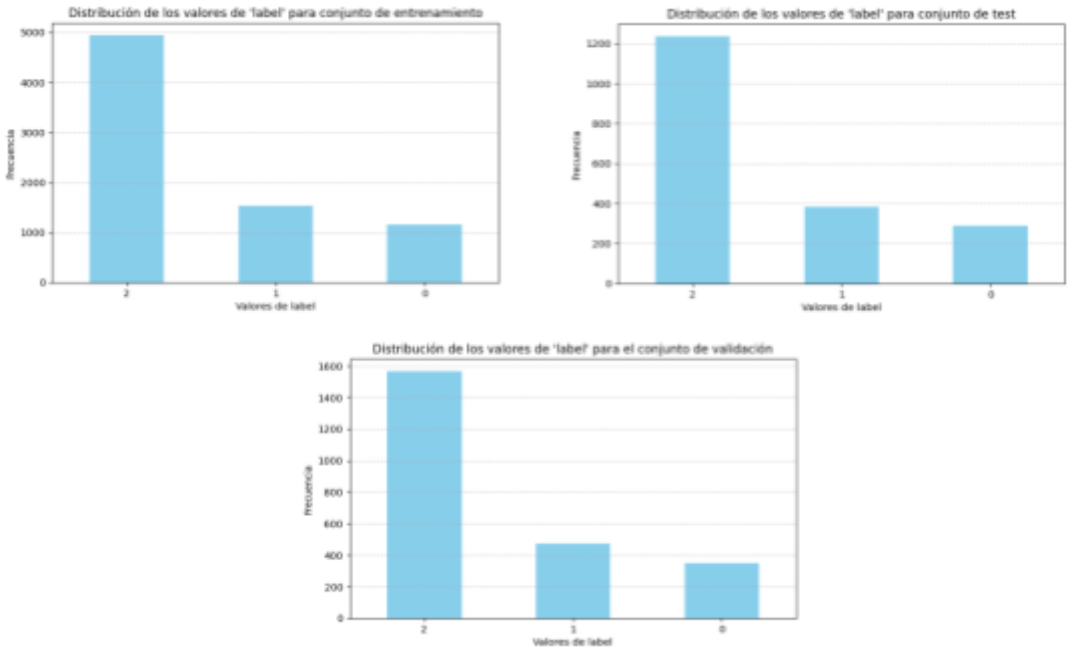


Figura 2

Distribución de los valores de la columna “label” de los tres conjuntos de datos



Una vez divididos los datos, se construyeron tres modelos de redes neuronales para la clasificación de los tuits; todos los modelos fueron entrenados durante 20 o 30 periodos con un tamaño de lote fijo de 32. El primer modelo fue una red neuronal recurrente (RNN), el segundo una red neuronal convolucional (CNN), y el tercero un modelo de perceptrón multicapa (MLP). Todas estas redes neuronales tienen las mismas configuraciones. Utilizan el optimizador Adam con una tasa de aprendizaje de 0.0001. La función de pérdida empleada fue entropía cruzada categórica dispersa, y como métrica de evaluación se utilizó la precisión (accuracy).

Para evitar el sobreajuste y mejorar la convergencia, se implementaron dos técnicas de regularización: EarlyStopping y ReduceLROnPlateau. EarlyStopping monitoreó la pérdida de validación (val_loss) y detuvo el entrenamiento si no se observaba una mejora en los últimos 5 períodos. Por otro lado, ReduceLROnPlateau redujo la tasa de aprendizaje a la mitad si la pérdida de validación no mejoraba en los últimos 2 períodos, con un valor mínimo de tasa de aprendizaje de 1e-6. Es relevante mencionar que para fijar estos parámetros primero se realizaron pruebas ajustando cada uno de ellos, se comparó el optimizador Adam con SGD variando la tasa de aprendizaje en valores que se encontraban en el rango de 0.01 a 0.00001, pero para todos los modelos se halló que el optimizador Adam con una tasa de aprendizaje de 0.0001 daba los mejores resultados.

El primer modelo consistió en una red neuronal recurrente de 9 capas. La primera capa fue un codificador, seguido de una capa de incrustación que transformó las entradas en vectores densos de 128 dimensiones. La tercera capa fue una capa LSTM bidireccional con 128 neuronas, configurada para devolver secuencias, y con regularización L2. La cuarta capa fue una capa de normalización por lotes, seguida de una segunda capa LSTM bidireccional con 64 neuronas, configurada para devolver un vector único, con regularización L2. La sexta capa fue otra capa de normalización por lotes, seguida de una capa densa con 64 neuronas, con una función de activación ReLU y regularización L2. La octava capa fue una capa de Dropout con una tasa del 50%. La capa final utilizó activación softmax para clasificar en tres etiquetas.

El segundo modelo consistió en una red neuronal convolucional de 9 capas. La primera capa fue un codificador que transformó las entradas en números enteros, seguido de una capa de incrustación para convertir las entradas en vectores densos de 128 dimensiones. La tercera capa fue una capa convolucional con 64 filtros de tamaño 5x5, con una función de activación ReLU y regularización L2, seguida de una capa de normalización por lotes. La quinta capa fue una capa de Max Pooling para reducir la dimensionalidad, seguida de una capa densa con 32 neuronas, una función de activación ReLU y regularización L2. La séptima capa fue una capa de Dropout con una tasa del 70%, seguida de una capa de normalización por lotes. Finalmente, la capa de salida utilizó activación softmax para clasificar las tres etiquetas.

El tercer modelo consistió en una red de perceptrones multicapa (MLP) de 10 capas. La primera capa fue una capa de entrada para recibir secuencias. La segunda capa fue una capa de incrustación que representó las entradas como vectores densos de 128 dimensiones, seguida de una capa Flatten para aplanar las representaciones. La cuarta capa fue una capa densa con 128 neuronas, con una función de activación ReLU, seguida de una capa de normalización por lotes. La sexta capa fue una capa de Dropout con una tasa del 50%. La séptima capa fue otra capa densa con 64 neuronas y activación ReLU, seguida de una capa de normalización por lotes y otra capa de Dropout con una tasa del 70%. La capa final utilizó activación softmax para clasificar las tres etiquetas.

Para definir un modelo final, se compararon los valores de F1 ponderado calculado para cada uno de los modelos, utilizando las matrices de confusión generadas utilizando el conjunto de datos de test. Se decidió utilizar el F1 ya que aunque el conjunto de datos de entrenamiento ya no presentaba problemas de desbalance de clases el conjunto de test si y está métrica es más apropiada en estos casos.

Para el desarrollo del análisis, se utilizó Python (Van Rossum & Drake, 2009) en la plataforma Google Colab (Google Research, 2023), con las bibliotecas tensorflow (Abadi et al., 2016), pandas (McKinney, 2010), numpy (Harris et al., 2020), nltk (Bird et al., 2009), wordnet (Miller, 1995), sklearn (Pedregosa et al., 2011) y matplotlib (Hunter, 2007).

RESULTADOS

La red neuronal recurrente (RNN) entrenada tuvo problemas de generalización a datos no vistos durante el entrenamiento, esto puede ser observado en la figura 3 donde el accuracy oscila mucho y en realidad no llega a superar el 80%, aunque se debe recordar que no es la métrica más confiable en problemas de desbalance de clases, como se tiene en el conjunto de validación y es por eso que para poder evaluar si se trataba de un problema de sobreajuste del modelo la evaluación se centró en el monitoreo de la función de pérdida la cual como se mencionó anteriormente era la entropía cruzada categórica dispersa, función que no se ve tan afectada por el desbalance de clases como el accuracy, en la figura 4 se puede apreciar el comportamiento de esta, la cual después del epoch o época 10 parece no mejorar más. Es por esto que fue importante la implementación del early stopping, que recupera los pesos de la red en ese momento donde ya no hay más mejoras.

Figura 3

Resultados de accuracy según época para la red neuronal recurrente

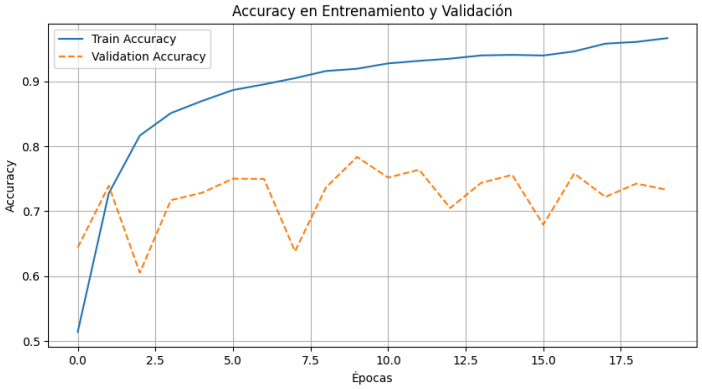
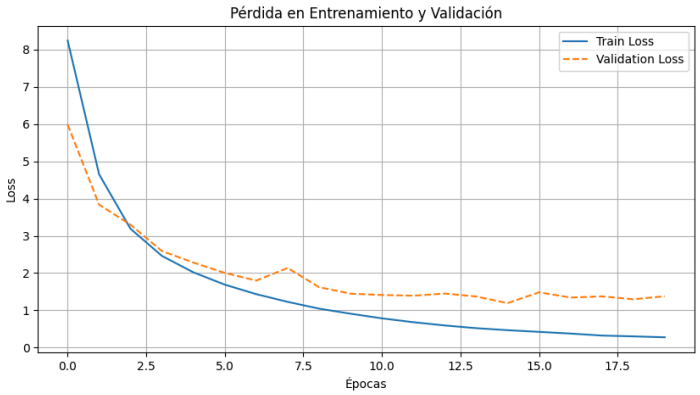


Figura 4

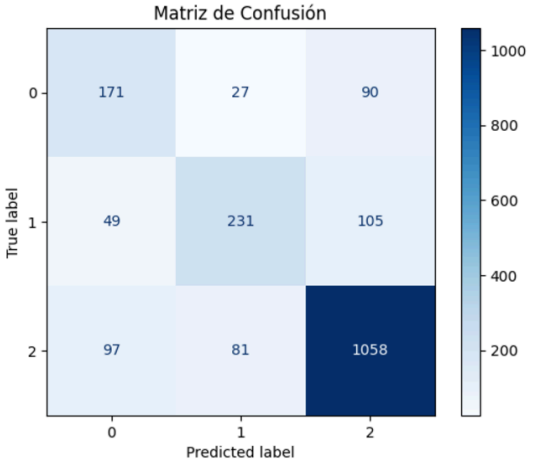
Resultados de pérdida según época para la red neuronal recurrente



Con los pesos de la época o epoch donde ya no había más mejora en la función de pérdida del modelo se realizó una matriz de confusión (figura 5), prediciendo valores para cada categoría utilizando los datos del conjunto de test, en este se observa que para las 3 categorías las clasificaciones son aceptables, ya que no hay tanta confusión entre las categorías 1 y 0 que son las que podrían representar afectaciones al rendimiento de un portafolio de inversiones siendo las categorías positiva y negativa respectivamente.

Figura 5

Matriz de confusión con datos de test para la red neuronal recurrente



Después la red neuronal convolucional tiene oscilaciones menores en el accuracy del conjunto de validación (figura 6) pero este accuracy sigue siendo parecido al del mejor modelo o los mejores pesos guardados de la red neuronal recurrente, aunque esta red aparentemente tuvo mejores rendimientos de la función de pérdida (figura 7) con mejoras constantes durante más períodos que la red neuronal recurrente.

Figura 6

Resultados de accuracy según época para la red neuronal convolucional

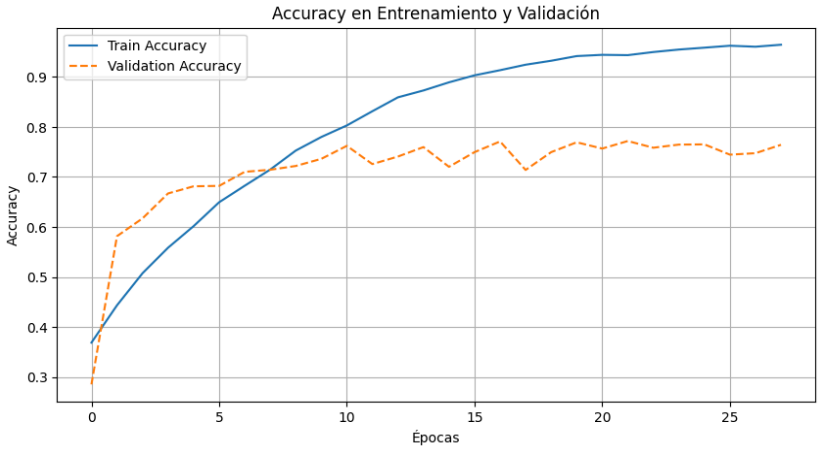
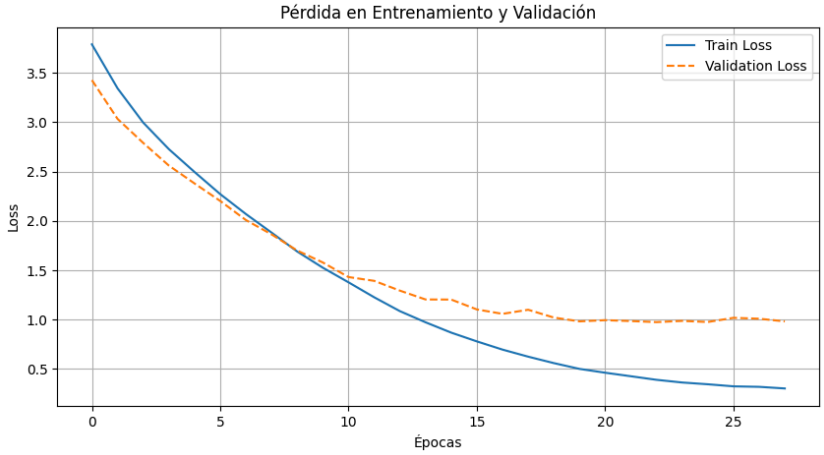


Figura 7

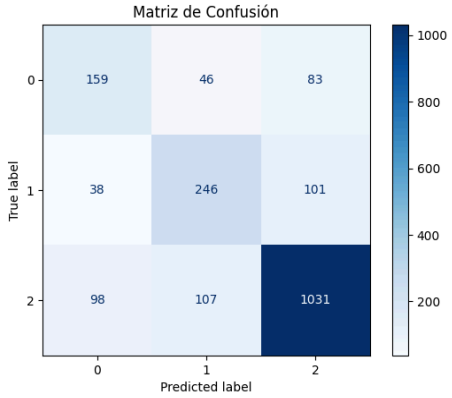
Resultados de pérdida según época para la red neuronal convolucional



Observando la matriz de confusión de la red neuronal convolucional (figura 8) se aprecia a simple vista que aparentemente posee un rendimiento muy parecido entre las clases 0 y 1, que, como ya se comentó anteriormente, son las de mayor impacto para decisiones relacionadas a inversiones, a la red neuronal recurrente.

Figura 8

Matriz de confusión con datos de test para la red neuronal convolucional



Finalmente el perceptrón multicapa parece tener un rendimiento similar a la red neuronal convolucional en accuracy (figura 9), aunque el rendimiento de la pérdida (figura 10) fuera mucho más oscilante y con mejoras menos relevantes que en las redes anteriores.

Figura 9

Resultados de accuracy según época para el perceptrón multicapa

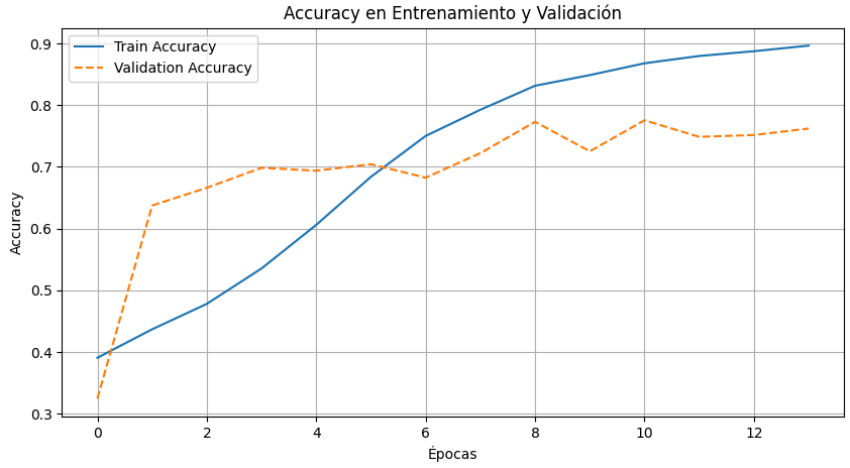
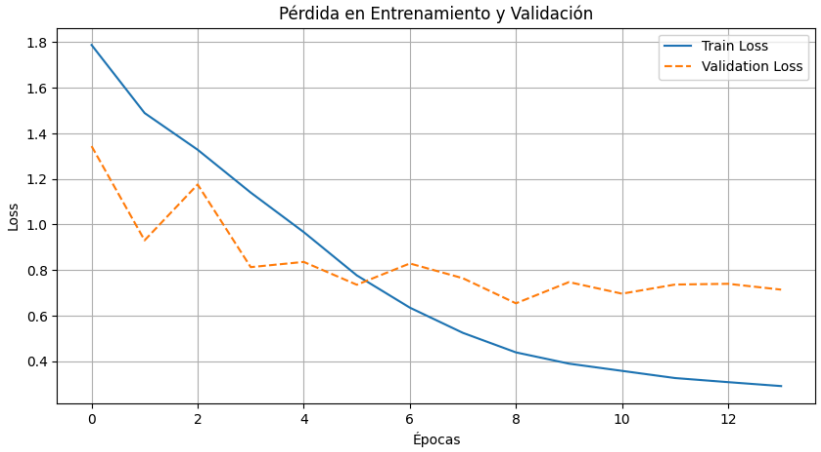


Figura 10

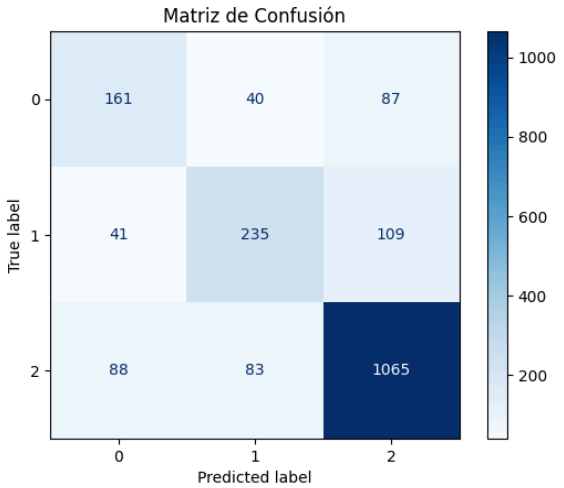
Resultados de pérdida según época para el perceptrón multicapa



Viendo la matriz de confusión del perceptrón multicapa (figura 11) se observó un rendimiento muy similar al de las dos redes anteriores, aunque a simple vista se podría considerar ligeramente mejor la red neuronal recurrente.

Figura 11

Matriz de confusión con datos de test para el perceptrón multicapa



Al no obtener una opción de modelo final clara solamente observando los resultados de clasificación se procedió a calcular el F1-score de cada uno de estos modelos. Al examinar los F1 por clase y el total ponderado, se observa que los tres modelos presentan métricas muy similares. Destacando que las clases de Bearish y Bullish tienen valores significativamente menores a comparación con la clase Neutral. Esta situación se debe al desbalance de clases presentes en el conjunto de test.

Cuadro 1

F1 Score por modelo

Modelo	F1 Score			
	Bearish	Bullish	Neutral	Total
RNN	0.565	0.638	0.850	0.7644
CNN	0.545	0.628	0.841	0.7536
MLP	0.557	0.633	0.853	0.7639

CONCLUSIONES

En conclusión, todos los modelos evaluados obtuvieron resultados similares, pero se seleccionó finalmente la RNN debido a que mostró métricas ligeramente superiores en las dos categorías consideradas más importantes: *Bearish* y *Bullish*. Además, se tomó en cuenta la naturaleza de los datos y el momento de su recolección. Dado que la mayoría de los tuits, si no es que todos, fueron generados antes de aproximadamente 2021, estaban limitados a 280 caracteres. Esto representa una ventaja para arquitecturas como la MLP, que procesan secuencias de longitud fija, especialmente considerando que esta longitud no representa una secuencia particularmente larga. Sin embargo, con la reciente introducción de Twitter Blue y la posibilidad de publicar tuits de hasta 4000 caracteres, se considera que la RNN ofrece una mejor capacidad de generalización frente a estos nuevos casos.

En general, para todos los modelos evaluados, el optimizador Adam con una tasa de aprendizaje de 0.0001 resultó ser el más efectivo, aunque la comparación se limitó al optimizador SGD con tasas de aprendizaje que variaron entre 0.01 y 0.00001. Adicionalmente, se reconoce el potencial de explorar modelos basados en Transformers, ya que las secuencias de 4000 caracteres podrían ser demasiado extensas incluso para modelos como la RNN.

Al comparar con otros trabajos del mismo tópico se encontró que por lo general los modelos se suelen centrar en clasificación binaria, es decir solo sentimientos positivos o negativos sobre el comportamiento de la acción de la que habla la noticia o en este caso el tuit. Este es el caso de Andrey Shtrauss (2024) quien se centra solo en estos dos sentimientos para analizar noticias financieras y crear una estrategia de trading basado en esto, argumentando que en realidad un sentimiento neutral no implica una acción y que esto podría ya estar considerado con solamente dos sentimientos, por ejemplo en el caso de comprar una acción nueva, un sentimiento neutral diría que no la compre, que es el mismo caso para el sentimiento negativo o en el caso de querer ver si mantener su posición en las acciones de una compañía o no, un sentimiento neutral indicaría mantener la posición, al igual que lo haría un sentimiento positivo.

En este análisis, se observa que la clase "neutral" es la principal causa del desbalance de clases en los datos, lo que afecta el rendimiento general de los modelos. Al haber una cantidad mayor de enunciados neutrales en comparación con los positivos o negativos, los modelos tienden a sobreajustarse a esta clase. Por lo tanto, se propone como posible mejora futura la eliminación de esta clase para centrar los modelos únicamente en clasificaciones binarias. Este enfoque podría simplificar el problema, reducir el desbalance y, potencialmente, mejorar la precisión y generalización de los modelos.

REFERENCIAS

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Zheng, X. (2016). TensorFlow: A system for large-scale machine learning. In Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (pp. 265-283).

Bird, S., Klein, E., & Loper, E. (2009). *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O'Reilly Media. https://www.nltk.org/

Chan, W. s. (2003). Stock price reaction to news and no-news: Drift and reversal after headlines. Journal of Financial Economics, 70(2), pp. 223–260.

Esrat Maria. (2020). *Data Augmentation with NLP* [Repositorio en GitHub]. GitHub. https://github.com/EsratMaria/Data_Augmentation_with_NLP/blob/master/Data%20Augmentation%20NLP.ipynb

Google Research. (2023). Google Colaboratory. https://colab.research.google.com/

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., ... & Oliphant, T. E. (2020). Array programming with NumPy. Nature, 585(7825), 357-362. https://doi.org/10.1038/s41586-020-2649-2

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. Computing in Science & Engineering, 9(3), 90-95.

Hugging Face. (2022). Twitter Financial News. https://huggingface.co/datasets/zeroshot/twitter-financial-news-sentiment

McKinney, W. (2010). Data structures for statistical computing in Python. In S. van der Walt & J. Millman (Eds.), *Proceedings of the 9th Python in Science Conference* (pp. 51–56). https://doi.org/10.25080/Majora-92bf1922-00a

Miller, G. A. (1995). WordNet: A lexical database for English. *Communications of the ACM, 38*(11), 39–41. https://doi.org/10.1145/219717.219748

Mostafa, M. M. (2013). More than words: Social networks' text mining for consumer brand sentiments. Expert Systems with Applications, 40(10), 4241–4251. https://doi.org/10.1016/j.eswa.2013.01.019

Schoen, h., gayo-avello, d., Metaxas, P. t., MustaFaraJ, e., strohMaIer, M., y Gloor, P. (2013). The power of prediction with social media.

Shtrauss, A (2024). *News sentiment-based trading strategy* [Notebook]. Kaggle. Retrieved November 24, 2024, from https://www.kaggle.com/code/shtrausslearning/news-sentiment-based-trading-strategy

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research, 12*, 2825–2830. https://scikit-learn.org/

Zagal, J. P., Tomuro, N., & Shepitsen, A. (2012). Natural language processing in game studies research: An overview. Simulation & Gaming, 43(3), 356–373. https://doi.org/10.1177/1046878111422560